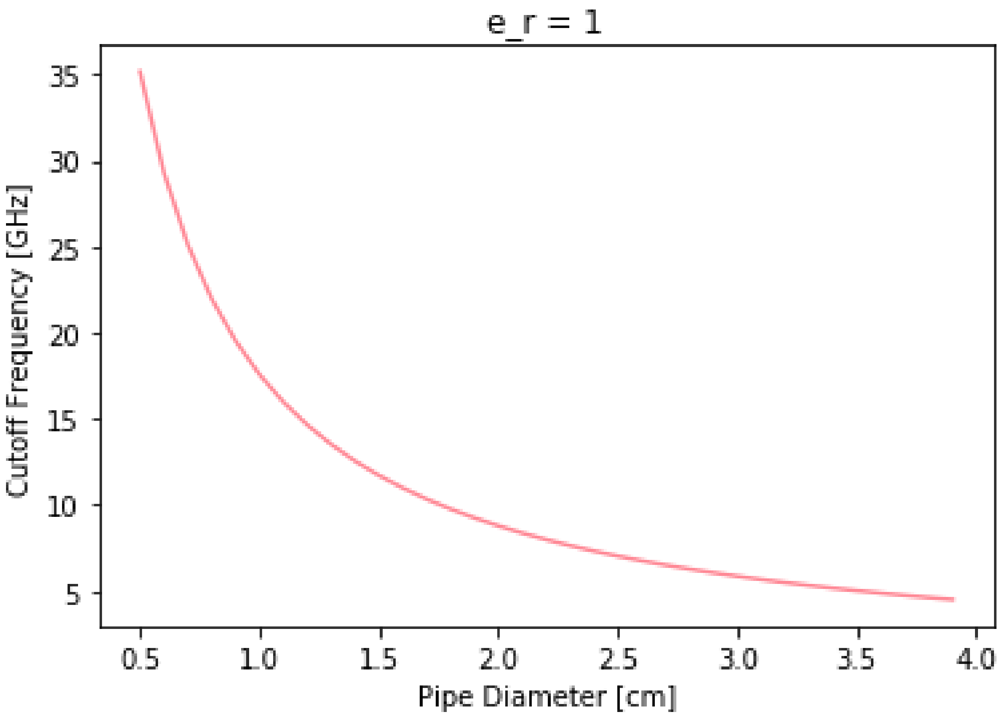
waveguide_graphs

April 2, 2018

```
In [23]: %matplotlib inline

         import pandas as pd
         import numpy as np
         import matplotlib

         from matplotlib import pyplot as plt

In [24]: # Constants
         c = 3e8
         f_c = 12e9
         # relative permittivity
         e_r = 1

In [25]: # Minimum pipe diameter based on the cutoff frequency
         d = (np.arange(0.5,4,.1))/100.
         fc = (1.841*c)/(2.0*np.pi*(d/2.0)*np.sqrt(e_r))
         for i in range(0,len(d),5):
             print("Diam=%.1f, cut_off=%.2f GHz"%(d[i]*100., fc[i]/1e9))

         print("Diam=%.2f, cut_off=%.2f GHz"%(.0127*100., ((1.841*c)/(2.0*np.pi*(.0127/2.0)*np.s
```

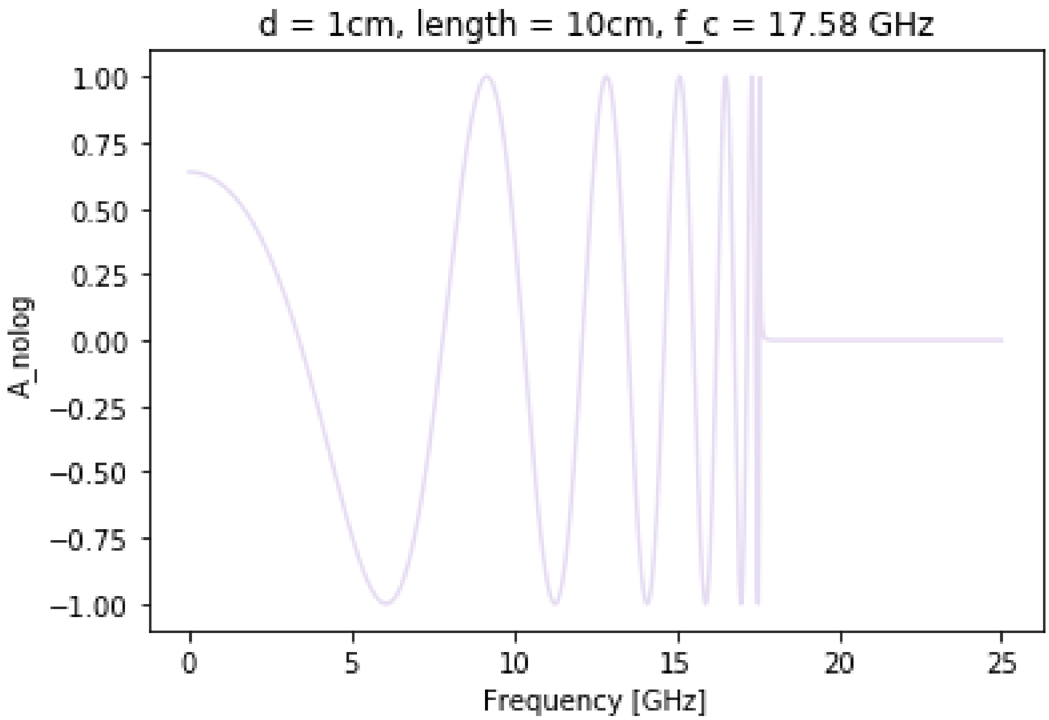
```
Diam=0.5, cut_off=35.16 GHz
Diam=1.0, cut_off=17.58 GHz
Diam=1.5, cut_off=11.72 GHz
Diam=2.0, cut_off=8.79 GHz
Diam=2.5, cut_off=7.03 GHz
Diam=3.0, cut_off=5.86 GHz
Diam=3.5, cut_off=5.02 GHz
Diam=1.27, cut_off=13.84 GHz
```

```
In [26]: #plt.subplot(211)
         plt.plot(d*100.0,(fc/1e9),'#ff8895')
         plt.xlabel('Pipe Diameter [cm]')
         plt.ylabel('Cutoff Frequency [GHz]')
         plt.title('e_r = 1')
         plt.show()
```

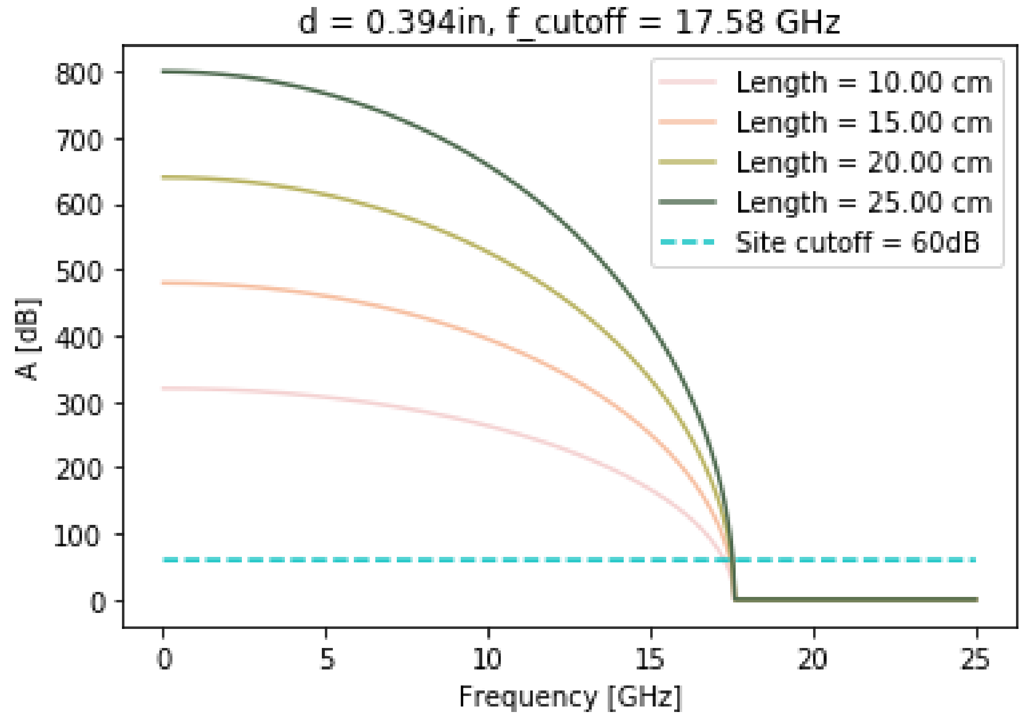


In [ ]:

In [27]:
```
length = .1
d = .01
f = (np.arange(9e6,25e9,1e5,dtype=np.complex))
#print(len(f))
k = ((2*np.pi*f)/c)
#print(k)
k_c = np.complex(1.841/(d/2.0))
b = np.sqrt(k**2-k_c**2)
A_nolog = np.exp(-b*length)

plt.plot(f.real/1e9, A_nolog.real,'#e6dcf2')
plt.xlabel('Frequency [GHz]')
plt.ylabel('A_nolog')
plt.title('d = 1cm, length = 10cm, f_c = 17.58 GHz')
plt.show()
```

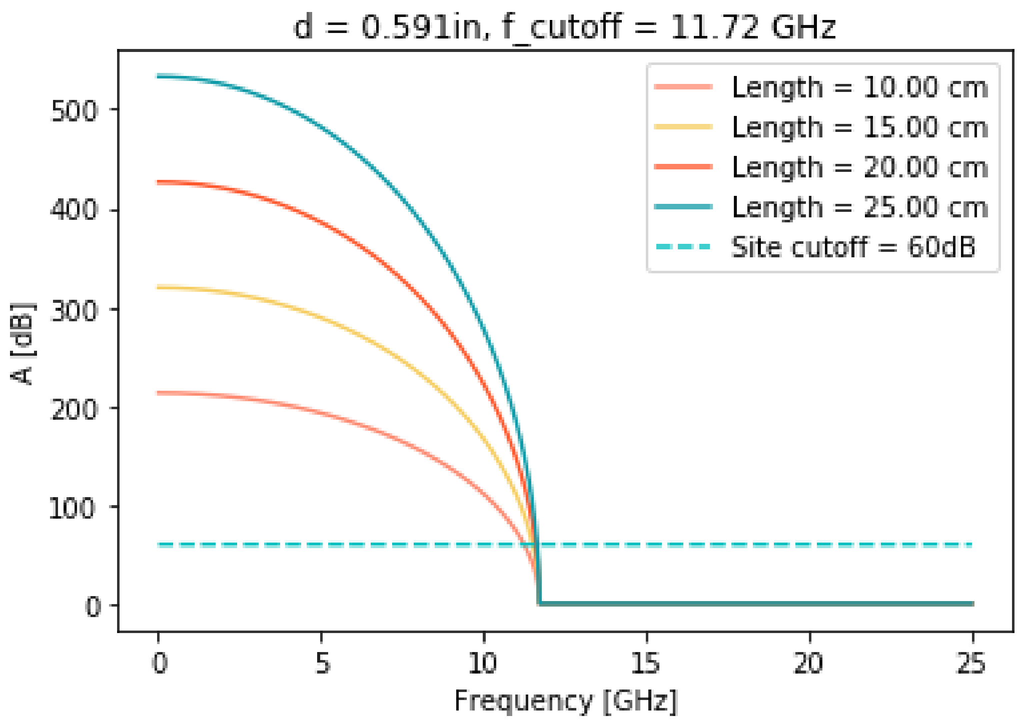



```
In [28]: lengths = [.1,.15,.2,.25]
         d = .01
         f = (np.arange(9e6,25e9,1e5,dtype=np.complex))
         k = ((2*np.pi*f)/c)
         k_c = np.complex(1.841/(d/2.0))
         b = np.sqrt(k**2-k_c**2)
         site_cutoff = np.ones(len(f))*60
         colors = ['#f4d2d2','#f6bfa1','#b8b260','#4b674c']


         for i,length in enumerate(lengths):
             A = -20*np.log10(np.exp(-b.imag*length))
             label = 'Length = %.2f cm'%(length*100)
             plt.plot(f.real/1e9,A,colors[i],label=label)
             plt.xlabel('Frequency [GHz]')
             plt.ylabel('A [dB]')
             plt.title('d = %.3fin, f_cutoff = 17.58 GHz'%(d*100/2.54))

         plt.plot(f.real/1e9,site_cutoff,'--c',label="Site cutoff = 60dB")
         plt.legend()
         plt.show()
```



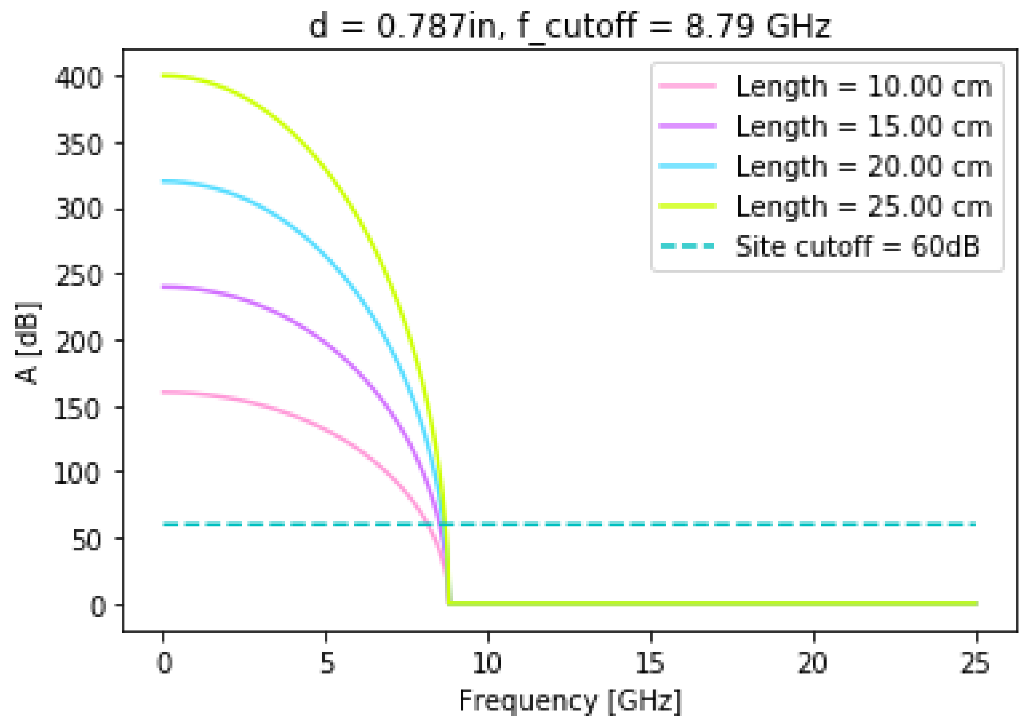
In [29]:
```
lengths = [.1,.15,.2,.25]
d = .015
f = (np.arange(9e6,25e9,1e5,dtype=np.complex))
k = ((2*np.pi*f)/c)
k_c = np.complex(1.841/(d/2.0))
b = np.sqrt(k**2-k_c**2)
site_cutoff = np.ones(len(f))*60
colors = ['#fe8a71','#f6cd61','#ff582d','#0e9aa7']


for i,length in enumerate(lengths):
    A = -20*np.log10(np.exp(-b.imag*length))
    print(i,length)
    label = 'Length = %.2f cm'%(length*100)
    print(label)
    plt.plot(f.real/1e9, A,colors[i],label=label)
    plt.xlabel('Frequency [GHz]')
    plt.ylabel('A [dB]')
    plt.title('d = %.3fin, f_cutoff = 11.72 GHz'%(d*100/2.54))

plt.plot(f.real/1e9,site_cutoff,'--c',label="Site cutoff = 60dB")
plt.legend()
plt.show()
```

```
0 0.1
Length = 10.00 cm
1 0.15
Length = 15.00 cm
2 0.2
Length = 20.00 cm
3 0.25
Length = 25.00 cm
```



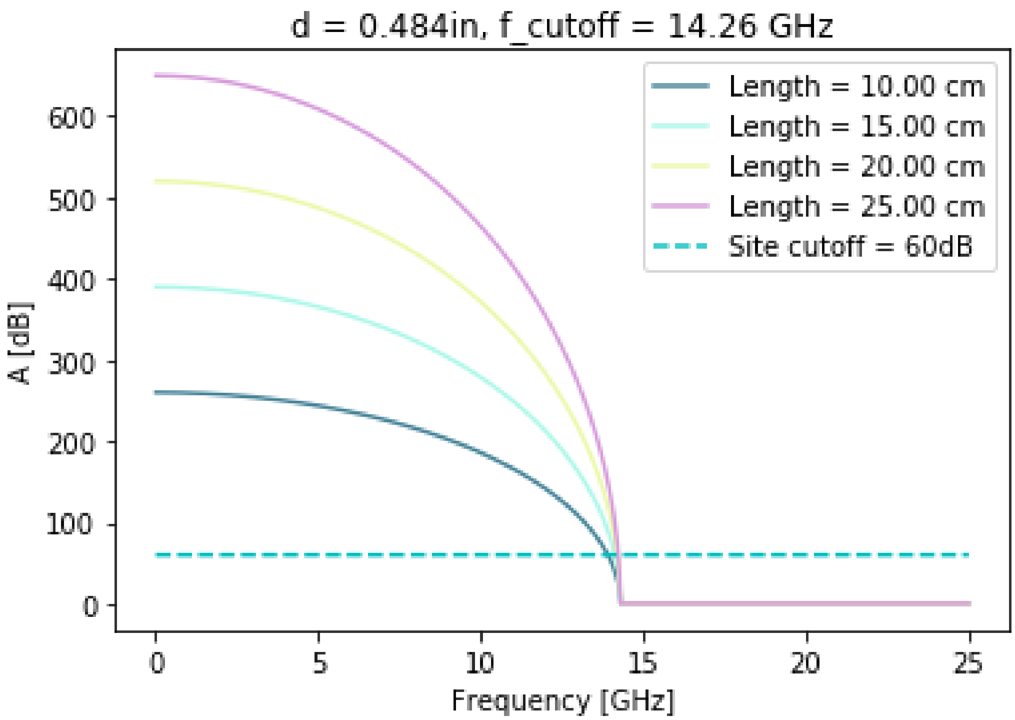

```
In [30]: lengths = [.1,.15,.2,.25]
         d = .020
         f = (np.arange(9e6,25e9,1e5,dtype=np.complex))
         k = ((2*np.pi*f)/c)
         k_c = np.complex(1.841/(d/2.0))
         b = np.sqrt(k**2-k_c**2)
         site_cutoff = np.ones(len(f))*60
         colors = ['#ff99d8','#d270ff','#52dcff','#ccff00']


         for i,length in enumerate(lengths):
             A = -20*np.log10(np.exp(-b.imag*length))
             label = 'Length = %.2f cm'%(length*100)
             plt.plot(f.real/1e9, A, colors[i], label=label)
```

```
    plt.xlabel('Frequency [GHz]')
    plt.ylabel('A [dB]')
    plt.title('d = %.3fin, f_cutoff = 8.79 GHz'%(d*100/2.54))

plt.plot((f/1e9).real,site_cutoff,'--c',label="Site cutoff = 60dB")
plt.legend()
plt.show()
```



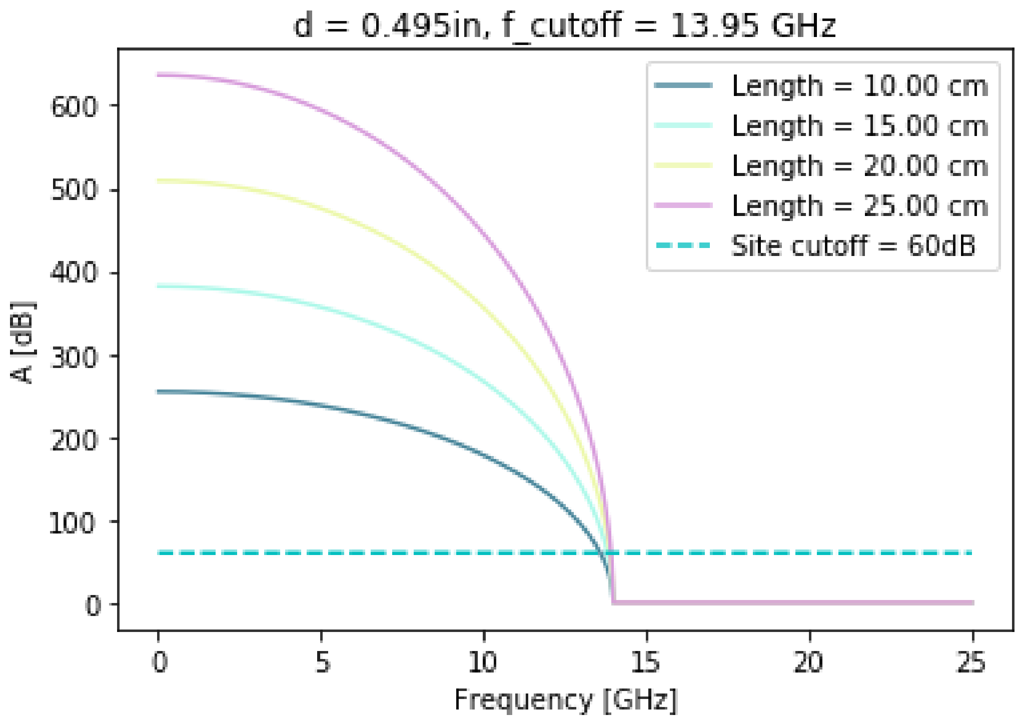

```
In [31]: lengths = [.1,.15,.2,.25]
         d = .0123
         f = (np.arange(9e6,25e9,1e5,dtype=np.complex))
         k = ((2*np.pi*f)/c)
         k_c = np.complex(1.841/(d/2.0))
         b = np.sqrt(k**2-k_c**2)
         site_cutoff = np.ones(len(f))*60
         colors = ['#468499','#acf8e9','#ecf8a9','#d79bdb']


         for i,length in enumerate(lengths):
             A = -20*np.log10(np.exp(-b.imag*length))
             label = 'Length = %.2f cm'%(length*100)
             plt.plot(f.real/1e9, A, colors[i], label=label)
             plt.xlabel('Frequency [GHz]')
             plt.ylabel('A [dB]')
```

```
    plt.title('d = %.3fin, f_cutoff = 14.26 GHz'%(d*100/2.54))

plt.plot((f/1e9).real,site_cutoff,'--c',label="Site cutoff = 60dB")
plt.legend()
plt.show()
```




```
In [32]: lengths = [.1,.15,.2,.25]
         d = .01257
         f = (np.arange(9e6,25e9,1e5,dtype=np.complex))
         k = ((2*np.pi*f)/c)
         k_c = np.complex(1.841/(d/2.0))
         b = np.sqrt(k**2-k_c**2)
         site_cutoff = np.ones(len(f))*60
         colors = ['#468499','#acf8e9','#ecf8a9','#d79bdb']


         for i,length in enumerate(lengths):
             A = -20*np.log10(np.exp(-b.imag*length))
             label = 'Length = %.2f cm'%(length*100)
             plt.plot(f.real/1e9, A, colors[i], label=label)
             plt.xlabel('Frequency [GHz]')
             plt.ylabel('A [dB]')
             plt.title('d = %.3fin, f_cutoff = 13.95 GHz'%(d*100/2.54))
```

```
plt.plot((f/1e9).real,site_cutoff,'--c',label="Site cutoff = 60dB")
plt.legend()
plt.show()
```



In [33]:
```
# The waveguide inner diameter of 0.495 inches was initially called out
# in the drawings. SSL manufactured the waveguides at
# the inner diameter of 0.500 inches because it was easier
# to machine that way. The prototype node has waveguides at
# inner diameter of 0.500 inches which brings the cutoff down to 13.84 GHz
```

In [ ]: